[예제 8-1] ex08-01.c

```
#include <sys/types.h>
#include <unistd.h>

main()
{
    pid_t pid;
    int status;

    pid = fork();

    putenv("APPLE=RED");

    if(pid > 0)
    {
        printf("[parent] PID : %d\n", getpid());
        printf("[parent] PPID: %d\n", getppid());
        printf("[parent] GID : %d\n", getpgrp());
        printf("[parent] SID : %d\n", getsid(0));

        waitpid(pid, &status, 0);

        printf("[parent] status is %d\n", status);

        unsetenv("APPLE");
    }
    else if(pid == 0)
    {
        printf("[child] PID : %d\n", getpid());
        printf("[child] PPID: %d\n", getppid());
        printf("[child] GID : %d\n", getpgid(0));
        printf("[child] SID : %d\n", getsid(0));

        sleep(1);

        printf("[child] APPLE=%s\n", getenv("APPLE"));

        exit(1);
    }
```

```
    else
        printf("fail to fork\n");
}
```

[예제 8-2] ex08-02.c

```
#include <stdio.h>
#include <sys/types.h>

main()
{
    pid_t pid;
    int status;

    pid = fork();

    if(pid > 0)
    {
        printf("parent: waiting..\n");
        wait(&status);
        printf("parent: status is %d\n", status);
    }
    else if(pid == 0)
    {
        sleep(1);
        printf("child: bye!\n");
        exit(1234);
    }
    else
        printf("fail to fork\n");

    printf("bye!\n");
}
```

[예제 8-3] ex08-03.c

```
#include <unistd.h>
#include <sys/types.h>

main()
{
    pid_t pid1, pid2;
    int status;

    pid1 = pid2 = -1;

    pid1 = fork();
    if(pid1 > 0)
        pid2 = fork();

    if(pid1 > 0 && pid2 > 0)
    {
        waitpid(pid2, &status, 0);
        printf("parent: child2 - exit(%d)\n", status);
        waitpid(pid1, &status, 0);
        printf("parent: child1 - exit(%d)\n", status);
    }
    else if(pid1 == 0 && pid2 == -1)
    {
        sleep(1);
        exit(1);
    }
    else if(pid1 > 0 && pid2 == 0)
    {
        sleep(2);
        exit(2);
    }
    else
        printf("fail to fork\n");
}
```

[예제 8-4] ex08-04.c

```
#include <unistd.h>
#include <sys/types.h>
#include <sys/wait.h>

main()
{
    pid_t pid;
    int status = 0;

    if((pid = fork()) > 0)
    {
        while(!waitpid(pid, &status, WNOHANG))
        {
            printf("parent: %d\n", status++);
            sleep(1);
        }
        printf("parent: child - exit(%d)\n", status);
    }
    else if(pid == 0)
    {
        sleep(5);
        printf("bye!\n");
        exit(0);
    }
    else
        printf("fail to fork\n");
}
```

[예제 8-5] ex08-05.c

```
#include <sys/types.h>
#include <unistd.h>

main()
{
    pid_t pid;

    if((pid = fork()) > 0)
    {
        printf("[ex08-05.c] PPID:%d, PID:%d\n", getppid(), getpid());
        sleep(1);
    }
    else if(pid == 0)
    {
        printf("[ex08-05.c] PPID:%d, PID:%d\n", getppid(), getpid());
        execl("0806", "0806", (char *)0);
    }
    else
        printf("fail to fork\n");
}
```

[예제 8-6] ex08-06.c

```
#include <sys/types.h>
#include <unistd.h>

main()
{
    printf("[ex08-06.c] PPID:%d, PID:%d\n", getppid(), getpid());
}
```

[예제 8-7] ex08-07.c

```
#include <sys/types.h>
#include <unistd.h>

main()
{
    printf("getpgrp():%d\n", getpgrp());
    printf("getpgid(0):%d\n", getpgid(0));
    printf("getpgid(getpid()):%d\n", getpgid(getpid()));
}
```

[예제 8-8] ex08-08.c

```
#include <sys/types.h>
#include <unistd.h>

main(int argc, char *argv[])
{
    pid_t pid;
    int interval;

    if(argc != 3)
        exit(1);

    pid = atoi(argv[1]);
    interval = atoi(argv[2]);

    printf("shell process...\n");
    printf("process id:%d, group id:%d, session id:%d\n",
        pid, getpgid(pid), getsid(pid));
    printf("current process.. not daemon...\n");
    printf("process id:%d, group id:%d, session id:%d\n",
        getpid(), getpgrp(), getsid(0));

    sleep(interval);
}
```

[예제 8-9] ex08-09.c

```
#include <sys/types.h>
#include <unistd.h>

main()
{
        pid_t pid;

        if((pid = fork()) > 0)
        {
                sleep(1);
                exit(1);
        }
        else if(pid == 0)
        {
                printf("old session id: %d\n", getsid(0));
                printf("new session id: %d\n", setsid());
                sleep(600);
        }
}
```

[예제 8-10] ex08-10.c

```
#include <unistd.h>

main()
{
    putenv("APPLE=BANANA");
    printf("%s\n", getenv("APPLE"));

    execl("ex08-11", "ex08-11", (char *)0);
}
```

[예제 8-11] ex08-11.c

```
#include <unistd.h>

main()
{
    printf("%s\n", getenv("APPLE"));
    unsetenv("APPLE");

    if(!getenv("APPLE"))
        printf("APPLE not found\n");
}
```

[예제 8-12] ex08-12.c

```
#include <unistd.h>

main()
{
        char *envlist[] = {"APPLE=0", "BANANA=1", (char *)0 };

        execle("ex08-13", "ex08-13", (char *)0, envlist);
}
```

[예제 8-13] ex08-13.c

```
#include <unistd.h>

extern char **environ;

main()
{
        while(*environ)
                printf("%s\n", *environ++);
}
```